A Comprehensive Investigation of Distribution in the Context of Workflow Management

Stefan Jablonski, Ralf Schamburger,
Christian Hahn, Stefan Horn, Rainer Lay, Jens Neeb, Michael Schlundt

University of Erlangen-Nuremberg
Department of Computer Science VI
Martensstr. 3, D-91058 Erlangen, Germany
{jablonski, schamburger}@informatik.uni-erlangen.de

Abstract

Distribution is often discussed under different issues of interest in the context of workflow management. This paper contributes a general framework for this discussion, i.e. it presents a general framework for workflow management in distributed environments. Technical and organizational issues are analyzed that have an impact on the design of a distributed workflow management system. Within the core part of this paper we present a taxonomy enabling to compare implementation concepts of different workflow management systems especially with respect to distribution.

1. Motivation

"Distribution" and "workflow management" are topics which are naturally associated with each other closely. There are articles about 'distributed workflow management' [1], [2], 'wide area workflow management' [3], and 'Cross-Organizational Workflow Management and Co-ordination' [4] which all deal with the issue 'distribution'. These articles investigate architectures for workflow management systems deployable in distributed computer systems, techniques for enacting workflows in large scale computer networks whereby the interaction of different workflow management systems is discussed, and organizational issues of workflow management in distributed organizations. We wonder how all these contributions can be distinguished from each other and whether they are comparable, orthogonal and complementary.

We start out our investigations from the assumption that distribution and workflow management are inherently and intrinsically coupled. A workflow is always distributed, there is no centralized workflow; of course, it has to be defined what "distribution" means in this context. A brief look onto a definition of workflow management supports this assumption:

> *A Workflow management system is an active system that manages the flow of business processes performed by multiple persons. It gets the right data to the right people with the right tools at the right time [5].*

The main components of a workflow, namely persons, tools and data, will in general not be allocated at one single computer. Since a workflow has to integrate all these elements, the naturally distributed character of a workflow can be derived. The definitions indirectly imply that workflow management is performed on a distributed computer system. Besides, they convey that workflow defines processes that span different organizations and organizational units, respectively. Therefore, workflow management also reflects organizational concerns.

Now, we are able to rephrase our critical statements made above. "Distribution" as an intrinsical part of workflow management has to be investigated either from a technical perspective and from an organizational perspective. Moreover, since these perspectives mutually affect impact each other, a comprehensive and combined discussion of these issues is required. One consequence is to define a methodology that both is able to capture either technical and organizational parameters of an application and is able to derive a technical architecture from these parameters. This paper will contribute such a methodology.

Technical and organizational impacts on workflow management are discussed in Section 2. Section 3 introduces a general model for distributed systems, here applied to the realm of workflow management and structures the distribution problem finally. Related work is presented in form of table in Section 4. Enhancements to the workflow language for supporting and controlling distribution out of the workflow management application are presented in Section 5. The paper finishes with a sort conclusion.

2. Organizational and Technical Impacts on Workflow Management

These business processes distinguish themselves by integrating remote resources like organizational structures (e.g. users, groups, departments), applications (e.g. word processors, databases) and workunits (the tasks to perform). Workunits may thereby be seen as parts of a business process enacted by different organizational agents (e.g. departments, clerks). The distribution of a workflow management system is a natural outcome of the physical distribution of organizations, hardware devices, software applications and business logic. We assume that each organizational unit has got assigned a couple of hardware devices. Since users belong to one specific department they are automatically assigned to specific hardware components. Organizational units may even be located outside of the companies network. For instance, users work from home or working on a laptop while being on business trips.

Not only hardware devices are assigned to specific organizational units, but also (software) applications are. Nor-

0-7695-1153-8/01 $10.00 © 2001 IEEE

mally, applications are installed and made available on the computers of an organizational unit. Thus, their allocation is determined by organizational constraints. Another issue to take into consideration is the heterogeneity of the hardware and software infrastructure. Heterogeneity [6] is mostly caused by an autonomous and independent behavior of organizational units which select and buy hardware and software components as results of internal assessment. Thus, the natural outcome is that these components differ in 'syntax' (format, protocol) and semantics. Besides technical and organizational constraints that directly can be derived, some general principles must be obeyed. First of all security requirements must be fulfilled which reaches from privacy to authentication and authorization. Second, additional costs have to be taken into account. There must be rules that determine who is going to pay for what service during workflow execution. Finally, additional administration tasks for the distributed system must be considered.

It is not in the main scope of this paper to compile a complete list of technical and organizational constraints for a workflow management application. Instead, the above discussion demonstrates clearly that many different and probably sometimes contradictionary constraints have to be considered when a workflow management application is designed.

3. Distributed Workflow Application and Solution

Enacting a distributed business process is done by defining and executing an appropriate set of workflows, better workflow schemes. A workflow schema comprises workflow definitions, data definitions, agent definitions, application definitions, etc. The workflow schemes together constitute a workflow based application system (AS_{WfMS}). An AS_{WfMS} runs on top of a workflow management system (WfMS). Executing a workflow means to instantiate a workflow schema; this results in workflow instances. They are also regarded as part of the AS_{WfMS}. Derived from the term distributed system and application in the area of computer science (e.g. [7], cf. Fig. 1a), the terms distributed workflow application and solution can be introduced: A WfMS together with a AS_{WfMS} forms a distributed application, here called distributed workflow application (DWA). A distributed workflow solution (DWS) is built up by a distributed workflow application and its underlying hardware resources (cf. Fig. 1b).

In order to complete the discussion of distributed systems in the context of workflow management, the components A_i of a distributed application, here the DWA, still have to be identified. According to Figure 1b, either the AS_{WfMS} and the WfMS have to be analyzed with respect to components. Technical and organizational requirements towards workflow management are completely enacted through:

- the identification of components A_i, and
- the allocation of components A_i to hardware resources S_k.

Thus, this allocation reflects the impact of these requirements on workflow management. For instance, an organizational requirement forces the allocation of a WfMS at the department's host; another requirement leads to the replication of some components at different sites in order to increase their availability. Consequently, the identification of components A_i and their allocation to hardware resources (S_k), e.g. its distribution, is a direct consequence of organizational and technical issues. Note, due to this observation it is not required that a DWA must be distributed in an exhaustive manner. Also a mainly centralized architecture might fulfill the requirements accurately.

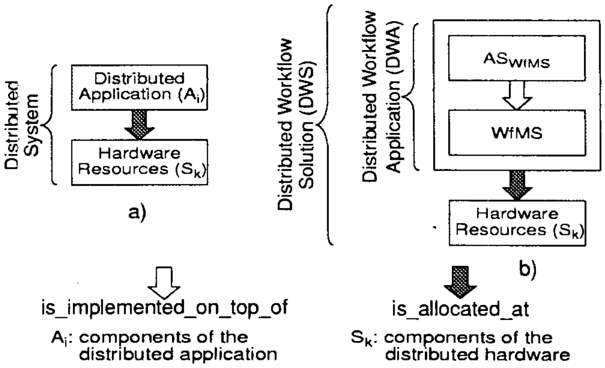
Distributed System
Distributed Application (A_i)
Hardware Resources (S_k)
a)
Distributed Workflow Solution (DWS)
Distributed Workflow Application (DWA)
AS_{WfMS}
WfMS
Hardware Resources (S_k)
b)
is_implemented_on_top_of
A_i: components of the distributed application
is_allocated_at
S_k: components of the distributed hardware

Fig. 1 Applying the definition of a distributed system to workflow management

Component Relationships

The primary task is to identify components. They belong either to the AS_{WfMS} (named $Comp_{AS}$) or to the WfMS ($Comp_{WfMS}$). The secondary task is to specify relationships between these components. These relationships determine the allocation of the components to hardware devices, i.e. the distribution of the DWS. The relationships are derived from technical and organizational constraints. A representative set of relationships is compiled into the following list (cf. Fig. 2):

(1) Hardware infrastructure: First of all, a description of the hardware infrastructure together with technical parameter is necessary. It is necessary to know about the available hardware components and the connectivity of the network between them.

(2) Relationships between organizational units: Organizational units depend on each other. For example, an organizational unit ou1 is a sub-department of ou2, an organizational unit (i.e. an employee) ou3 works for ou4 (e.g. a project). See [8] for an extensive discussion of this topic.

(3) Relationships between components and organizational units: Any of the components of figure 2 is assigned to organizational units (3a, 3b, 3c). People of these organizational units use the hardware devices to work with the computer system. Consequently, applications must be accessible from these hardware devices if they are going to be used.

(4) Relationships between $\text{Comp}_{\text{WfMS}}$ and hardware resources: A specific component $\text{Comp}_{\text{WfMS}}$ requires a certain hardware type. For instance, a certain workflow server runs on Solaris 2.7 operating system.

(5) Relationships among $\text{Comp}_{\text{WfMS}}$: The components of a WfMS mutually require each other according to client/server computing. These relationships especially have an impact on non-functional issues like and scalability.

(6) Relationships between $\text{Comp}_{\text{WfMS}}$ and Comp_{AS}: The components Comp_{AS} need selected components $\text{Comp}_{\text{WfMS}}$ to perform. For instance, the execution of an agent assignment procedure (as a Comp_{AS}) requires the corresponding module of a WfMS (i.e. a $\text{Comp}_{\text{WfMS}}$).

(7) Relationships among Comp_{AS}: For instance, during the execution of a workflow, agent and workflow type data are frequently needed together. Similar to the relationships (5), the allocation of Comp_{AS} thus has an important impact on non-functional requirements.

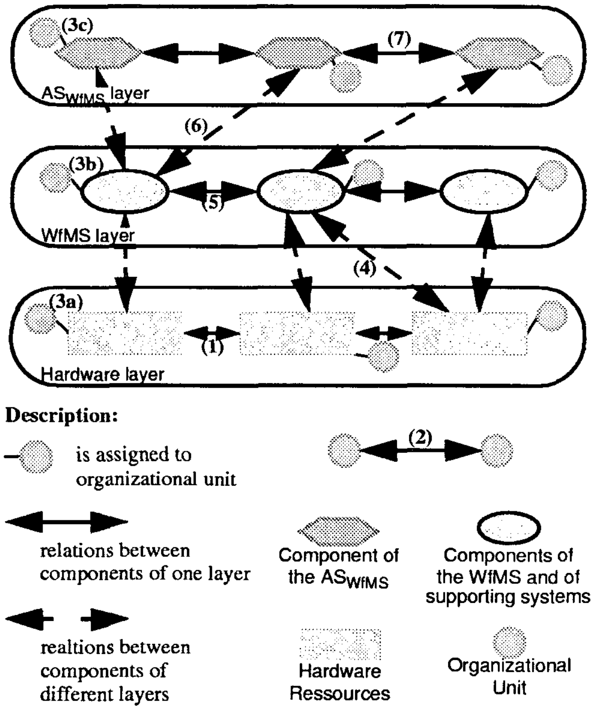
(3c)
(7)
AS$_{\text{WfMS}}$ layer
(6)
(3b)
(5)
WfMS layer
(4)
(3a)
(1)
Hardware layer
Description:
is assigned to organizational unit
(2)
relations between components of one layer
Component of the AS$_{\text{WfMS}}$
Components of the WfMS and of supporting systems
realtions between components of different layers
Hardware Ressources
Organizational Unit

Fig. 2 Components of the Distributed Workflow Solution and relations between them

Sources of Information

From an abstract point of view, the relationships depicted in the list above define the technical and organizational constraints that have a lasting impact on the distribution of a DWA. Consequently, it is necessary to know where these constraints are defined. In this sub-section we investigate the sources of information from that constraints are derivable.

The hardware infrastructure (list item (1)) is described in a system repository. Specifically for the use in the context of workflow management, the system repository must comprise information about connectivity, which is characterized by the bandwidth, latency and error rate of a network connection between two systems. Organizational units and their interrelationships (list item (2)) are defined in a so-called organizational model (cf. [16]). Such an organizational model is ideally independent of workflow management and is available for multiple application areas like project management, groupware and - of course - workflow management. In reality, an organizational model is mostly set up as a special module of a DWA. Workflow management systems support a language to define those organizational units and relationships that are required to find the right people to assign workflows to. The relationships between all kinds of components and organizational units (list item (3a, b, c)) are also part of a system repository. These relationships define the access rights of organizational units with respect to computer system and software resources and parts of the application.

Relationships between $\text{Comp}_{\text{WfMS}}$ and hardware resources (list item (4)), and among $\text{Comp}_{\text{WfMS}}$ (list item (5)) are defined in a kind of administration and installation guide of a WfMS. This guide specifies the degrees of freedom the implementors of a WfMS have incorporated into the system architecture. Here, it decides whether a WfMS can be implemented in a distributed way (e.g. as client/server system) or is more a monolithic system. Also, the degree of replication for a WfMS is set. Some WfMS can replicate functional components, some can replicate data stores, some can replicate both kinds of components. Altogether, list items (4) and (5) heavily determine the architecture of a WfMS. Vice versa, the architecture of a WfMS determines the technical and organizational constraints of type (4) and (5) that can be met by a concrete WfMS. This is way Section 6 discusses architectural issues of workflow management.

Technical and organizational constraints on a DWA are reflected by relationships (6) and (7). The associations between the components of a AS_{WfMS} and a WfMS are defined (list item (6)); furthermore, the interrelationships of the components of AS_{WfMS} are set (list item (7)). All related constraints are defined in a AS_{WfMS}. This implies, that the language of a AS_{WfMS} must provide for constructs to specify them. For example, the language must allow to specify that a concrete workflow (type) only may be executed on a certain WfMS or on a certain computer node. Another constraint forces the allocation of workflow data to that computer node where the workflow is instantiated. Constraints of these type have to be expressed in a workflow language. This language is needed to define workflows, policy resolution, etc., i.e. this language is needed to specify a AS_{WfMS}.

4. Related Work

This section discusses related work in the area of distribution in workflow management. The main goal is to compare the different approaches and see how they complement each other. As the basis for a comparison method we apply the taxonomy of relationships introduced in Section 3. This taxonomy facilitates to identify the main characteristics of

Table 1: Comparison of approaches for distribution in workflow management

System approaches	Focus	System distribution relationship (3,4)	Application distribution relationship (5,6)
CrossFlow [10]	Cross-Organisational Workflow by Contracts	- complete WfMS are assigned to organisations - distribution granularity: entire WfMS	- the application is distributed over several WfMS - distribution is specified in the workflow schemes and in contracts
Agent enhanced Workflow [11]	cross-organisational workflow	- coordination of several WfMS by agents - distribution granularity: entire WfMS	- workflow process is distributed over participating WfMS - execution plans are calculated by the agents based on goal definitions - distribution is described by execution plans
Event-Based Workflow Process Management [4]	distribution of process, multiple times zones, organisational boundaries and legal domains	- multiple WfMS are connected by an interoperability component - distribution granularity: entire WfMS	- relationships to model components of the workflow management application are not mentioned
ADEPT [12], [13]	enterprise wide workflows, cross enterprise workflows	- domains are hosted by workflow servers - distribution granularity: entire WfMS	- server assignments define assignment of activities to (workflow) servers - assignments are specified in workflow schema (language extension $\text{ADEPT}_{\text{distribution}}$)
Mentor [2]	scalable, enterprise wide workflow management	- autonomous workflow engines assigned to organizational units - distribution granularity: entire WfMS	- centralized workflow specification - partitioning by orthogonalization rules - assignment by rules (e.g. organizational assignment) to users and engines - rules are defined outside the workflow specification
Exotica/FMDC [1]	disconnected clients, mobile computing	- Runtime Client and Program Execution Client are separated from the Workflow Server - distribution granularity are built by these parts of the WfMS	- no specification about distribution in the workflow schema needed
Exotica/FMQM [14]	resilience, scalability, flexibility	- autonomous workflow system nodes cooperate for process execution - distribution granularity: entire WfMS	- process is partitioned and allocated to engines reflecting closeness to users and workload - allocation is based on users associated with different nodes - allocation is described outside a workflow schema
WISE [15]	virtual enterprises and business process	- distribution granularity: entire WfMS	- distribution is described in separated process model
WAWM [3]	inter-, intra organizational workflow	- uses shared coordination directories for multiple WfMS - distribution granularity: entire WfMS	- workflow application is distributed over autonomous WfMS - distribution is modelled in each workflow schema or in a separated virt. process definition
Mobile [9] with extensions see section 3 and 5	enterprise wide, cross organizational workflow	- distribution granularity: perspective servers of the WfMS	- AS_{WfMS} is partitioned and allocated to different WfMSs - distribution is modelled in the workflow schema - technical and organizational impacts are considered (section 5)

the related work approaches. Note, the relationships describe - directly and indirectly - technical and organizational constraints for workflow management.

However, the following investigation does not look further into the relationships (1) to (3) found in Section 3. We assume that these relationships are specified in all approaches since they build a basis for distribution. Thus, we concentrate on relationships (4) to (7), whereby the first two ((4) and (5)) deal with the distribution of a WfMS ($\text{Comp}_{\text{WfMS}}$) and the second two ((6) and (7)) with the distribution of a workflow management application (Comp_{AS}).

The results of the comparison can be summarized as follows. With the exception of Exotica/FMDC, the approaches deal with workflow distribution out of organizational reasons within and outside of a company. Three of them focus also on scalability and availability (ADEPT, Mentor, Exotica/FMWM).

The *distribution granularity* of $\text{Comp}_{\text{WfMS}}$ is always an entire workflow management system. Again, Exotica/FMDC is an exception; it splits the workflow management system into client/server parts to handle disconnected operations on the client side.That means that some functionality of the workflow engine is implemented at the client side. Also *MOBILE* is an exception; the workflow management system is spitted into different workflow perspective servers [9].

The distribution granularity of the components of a workflow management application (Comp_{AS}) are complete workflows. They are assigned to the distributed WfMS. One of our main interests is to find out where *partitioning* and *allocation* rules were defined. Three classes were found. The first class is characterized by not saying anything about distribution rules (Event-Based Workflow Process Management). The second class summarizes these approaches which define an individual definition language (CrossFlow, Agent enhanced Workflow, Mentor, Exotica/FMQM). This language is separated from the workflow specification language. The usage of a further language arises if the distribution goals and rules cannot be formulated in the given workflow language proper. The third class is formed by the approaches which extend the workflow language by distribution rules (CrossFlow, ADEPT, WISE, WAWM, *MOBILE*). The CrossFlow approach belongs to more than one class because the distribution rules are split into several definition parts. Contract definitions are defined separately because they do not fit into CrossFlow's workflow language.

Two different *distribution strategies* can be found in these approaches. The first strategy distributes the workflow management application independent of the application itself. There are a fixed set of rules which are coordinating the distribution. The second strategy enables the workflow application itself to directly influence the distribution. This strategy seems to be more flexible because the distribution can be dynamically adjusted to a concrete application scenario out of the application definition itself.

Although none of the given approaches allows for the combination of both strategies, this ought to be the best way to achieve utmost flexibility in distribution: Distribution for organizational reasons can be formulated in the workflow management application definition itself. Distribution for reasons of scalability and availability are application independent and should therefore be defined outside of the workflow application. Further investigation will be done within the scope of the *MOBILE* project.

5. A Workflow Language for the Specification of a Workflow based Application System

This section describes a language to define the relationships between components of the application system (Comp_{AS}) and components of the WfMS ($\text{Comp}_{\text{WfMS}}$). Referring to Section 3.3, list items (6) and (7) are discussed. What we want to ensure by these relationships is that workflows are performed on particular WfMS or components of a WfMS. There are two main reasons why these relationships are going to be specified. A first reason is security. For instance, a critical workflow based application should be executed at a WfMS behind a fire wall. Another reason is performance. For instance, it is known that a specific workflow is executed - more or less completely - within a particular organizational unit. Thus, it is recommended to perform it entirely on the WfMS assigned to this organizational unit. Further reasons like the heterogeneity of different WfMS are possible. We are not able to provide a comprehensive workflow language to specify arbitrary constraints in this section. Instead, we explain by two examples reason, principle structure and application of such a language.

Two questions have to be answered before we can continue. The first one refers to the specification of the relationships between the components. Where does it happen? We foster to specify these relationships in the workflow schemes. A workflow schema is aware of all Comp_{AS} and therefore is an ideal place. A second question is about how relationships are specified. Therefore, we are extending a workflow language [9] by assignment rules. These rules assign Comp_{AS} directly or indirectly to execution locations, i.e. hardware resources or $\text{Comp}_{\text{WfMS}}$. Two examples will illustrate the proceeding.

In our first example, we want to restrict the execution of the workflow *TravelClaimReimbursement* to the WfMS allocated in the administration department. This is a typical organizational constraint. The following language construct enforces this constraint:

```
ASSIGN TravelClaimReimbursement TO
          WfMS(administration_department)
```

Of course, also people sitting outside of this department are allowed to participate in this workflow. However, they only can access workflow data through a secure communication channel. The identification of this WfMS can be found by transitively browsing through a system repository.

Another example aims at performance. Here, a technical constraint has to be obeyed by the AS_{WfMS}. Figure 3 depicts a simplified car manufacturing workflow. Due to experience it is know that the three composite workflows Product Design, Parts Manufacturing and Parts Assembly are each executed at one specific location to a great extent. So it is not necessary that a specific workflow and data distribution module [17] calculates execution sites for these workflow parts. In the following, we present sections of a workflow definition where this constraint is specified:

```
WORKFLOW_TYPE CarManufacturing
  SUBWORKFLOWS ProductDesign,
          PartsManufacturing, PartAssembly;
  ASSIGN ProductDesign TO design_dep;
  ASSIGN PartsManufacturing TO subcontractor;
  ASSIGN PartAssembly TO assembly_dep;
  ...
END_WORKFLOW_TYPE
```

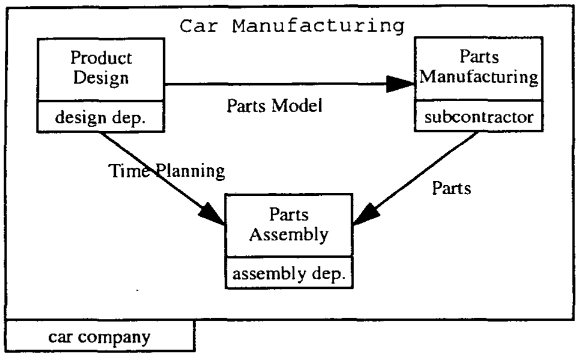

Fig. 3 Example Workflow: Car Manufacturing

As already said, this section just exposes some examples of typical language extensions to express technical and organizational constraints on workflow execution. In principle the constraints associate components of a workflow based application system, Comp_{AS}, to components of a workflow management system, Comp_{WfMS}. Such an assignment can be made dependent on organizational terms as can be derived from the above examples. In the first example, the WfMS of a specific organizational unit (i.e. administration_department) is specified as location; in the second example, organizational units (e.g. design_dep, subcontractor) are referenced. We are convinced that it is absolutely necessary and powerful to reference organizational units in assignment rules. This supports most flexible and individual assignment strategies. We are currently extending the workflow language of the MOBILE WfMS towards these language constraints.

6. Conclusion

This paper discusses distribution concepts in the area of workflow management. As a distributed system comprises distributed hardware and at least one distributed application we map this definition of distributed systems to distributed workflow management systems. We separate the discussion of distributed workflow management systems in both technical and organizational issues, that both have impact in design and implementation of such a system. Within the core part of this paper we present a taxonomy enabling to compare implementation concepts of different workflow management systems with respect to distribution.

7. References

[1] Alonso, G.; Günthör, R.; Mohan, C.; Kamath, M.; Agrawal, D.; El Abbadi, A.: *Exotica/FMDC: Handling disconnected clients in a workflow management system.* CoopIS-95, 1995.

[2] Muth, P.; Wodtke, D.; Weissenfels J.; Kotz Dittrich, A.; Weikum, G.: *From Centralized Workflow Specification to Distributed Workflow Execution.* In: JIIS - Special Issue on Workflow Managament, Volume 10, Number 2, Kluwer Academic Publishers, March 1998.

[3] Riempp, G.: *Wide Area Workflow Management.* Springer, London, 1998.

[4] Eder, J.; Panagos, E.: *Towards Distributed Workflow Process Management.* Workshop on Cross-Organisational Workflow Management and Co-ordination, together with WACC '99, San Francisco, February 1999.

[5] Reinwald, B.: *Workflow-Management in verteilten Systemen.* (in german), Teubner-Texte zur Informatik, Stuttgart, 1995.

[6] Wedekind, H.: *HEDAS: Heterogene, durchgängige Anwendungssysteme.* (in german), Abschlußbericht zum Sonderforschungsbereich (SFB) 182, Erlangen, 1999.

[7] Singhal, M.; Shivaratri, N.: *Advanced Concepts in Operating Systems.* McGraw-Hill, Inc., New York, 1994.

[8] Bussler, C.: *Policy Resolution in Workflow Management Systems.* Digital Technical Journal, Vol. 6, No. 4, Maynard, MA: Digital Equipment Corporation, 1995.

[9] Jablonski, S.; Bussler, C.: *Workflow Management - Modelling Concepts, Architecture and Implementation.* International Thomson Computer Press, September 1996.

[10] Ludwig, H.; Hoffner, Y.: *Contract-based Cross-Organisational Workflows - The CrossFlow Project.* In: Proceedings of the Workshop on Cross-Organisational Workflow Management and Co-ordination, San Francisco, 1999.

[11] Shepherdson, J.; Thompson, S.; Odgers, B.: *Cross Organisational Workflow Co-ordinated by Software Agents.* Intelligent Business Systems Research Gr., BT Laboratories, UK, 1999.

[12] Bauer, T.; Dadam, P.: *Enterprise-Wide and Cross-Enterprise Workflow-Management: Challenges and Research Issues for Adaptive Workflows.* CoopIS-97, Kiawah Island, 1997.

[13] Bauer, T.; Dadam, P.: *Efficient Distributed Control of Enterprise-Wide and Cross-Enterprise Workflows.* In: Proc. of the Workshop Informatik '99, GI, Paderborn 1999.

[14] Alonso, G.; Mohan, C.; Günthör, R.; Agrawal, D.; El Abbadi, A.; Kamath, M.: *Exotica/FMQM: A Persistent Message-Based Architecture for Distributed Workflow Management.* IFIP Conference, Trondheim, 1995.

[15] Lazcano, A.; Alonso, G.; Schuldt, H.; Schuler, C.: *The WISE approach to Electronic Commerce.* International Journal of Computer Systems Science & Engineering, 2000.

[16] Jablonski, S.; Bussler, C.: *Policy Resolution for Workflow Management Systems.* Proceedings of the Hawaii International Conference on System Sciences, 1995.

[17] Schuster, H.: *Architektur verteilter Workflow Management Systeme.* (in german), PhD-Thesis, Friedrich-Alexander-Universität, Erlangen, November 1997.